



## BAB II

### LANDASAN TEORI

#### A. Interface

Menurut Kamaruddin & Sulaiman, (2018) , Dalam membuat sebuah desain interface, ini sangatlah penting bagi desainer untuk mengerti kebutuhan dari para user, dimana sebelum dilakukannya pembuatan *interface*, dibutuhkan penelitian terlebih dahulu untuk jauh mengerti apa yang dibutuhkan oleh para pengguna atau user, dimana desainer melakukan akomodasi terhadap cara pandang user tentang kualitas dari konten yang dipresentasikan dalam suatu *interface*. Dimana seperti dengan yang dicantumkan oleh Kamaruddin & Sulaiman, (2018). mendefinisikan interface sebagai permukaan dari sebuah layar, yang memfasilitasi interpretasi bagaimana cara user melihat proses terjadinya komunikasi. Lalu, ada 4 definisi interface sesuai dengan yang dicantumkan Kamaruddin & Sulaiman, (2018):

1. Presentasi dari kontrol interface, dimana bagaimana cara user mencerna informasi ( melihat informasi tersebut)
2. Komunikasi kontrol dengan interface merupakan cara sistem berkomunikasi dengan user (sebuah metode komunikasi)
3. Navigasi interface mengontrol cara user melakukan navigasi dari bagian informasi yang satu ke yang lain ( pergerakan dari layar pertama ke layar yang lain)
4. Penjelasan dari kontrol interface, dimana user mengontrol aktivitas yang berbeda (penggunaan ikon).

#### 1. Karakteristik *interface*

Berikut di bawah ini adalah karakteristik *interface* menurut Kamaruddin & Sulaiman, (2018):



### a. Konsistensi

Di dalam analisis konten dalam literatur, konsistensi adalah hal yang paling sering dilihat sebagai prinsip yang penting di dalam desain *interface*, memfokuskan dalam kepentingan konsistensi penggunaan semua elemen *interface* seperti dengan font, jenis font, ukuran, warna, peletakan dari objek-objek seperti dengan gambar, judul, kata atau kalimat, suara, animasi dan navigasi sepanjang halaman dari sebuah konten multimedia untuk memfasilitaskan kemudahan penggunaan dan pemahaman user. Barfield , Kamaruddin , dan Wood memfokuskan bahwa objek-objek seperti teks, warna, grafik dan atau gambar, animasi, dan navigasi seharusnya diterapkan secara konsisten sepanjang desain *interface*, dengan tujuan mempermudah user ketika menelusuri desain *interface* dan pemahaman user selama melakukan navigasi.

### b. Hirarki

Prinsip dari hirarki, atau ketika kita lihat dalam teks, adalah hal kedua yang paling sering terjadi dalam prinsip desain *interface*. Hal ini diaplikasikan dengan cara melakukan kluster elemen *interface* seperti teks, gambar, dan grafik dengan memfokuskan tingkat kepentingan terhadap konten *interface*. Lalu menurut Johnson, hirarki akan mengikuti interaksi manusia dengan membatasi *redundancy* terhadap *cognitive load* terhadap user dalam lingkungan multimedia.

### c. Kontras

Dalam hal berhubungan dengan *legibility* dan *readability* sebagai “*Overall Screen-presentation*“, prinsip kontras, memiliki tingkat kepentingan yang sama dengan prinsip hirarki dan konsistensi, seperti yang sering

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



ditemukan dalam analisis konten. Kontras warna antara teks dengan *background* adalah hal yang paling sering didiskusikan oleh sarjanawan, dimana tujuan kontras adalah memfasilitasi *visibility* dan *readability* dari user. Sebagai contoh, menurut Williams dan Stimart merekomendasikan kontras dalam warna, teks, dan grafik digunakan untuk mencapai kejelasan dan mudah terbaca.

#### d. *Balance*

Telah diidentifikasi bahwa prinsip keseimbangan, seperti yang pernah terjadi, penting dalam memberikan pengalaman sehingga user mendapatkan pengalaman yang positif dengan mengaplikasikan keseimbangan dalam teks dan grafis sepanjang tata letak.

#### e. *Harmoni dalam desain interface*

Lalu untuk prinsip terakhir desain *interface* diidentifikasi sebagai yang paling sering terjadi, adalah harmoni atau dalam beberapa, yang terjadi dalam suatu kesatuan. Harmoni diaplikasikan dalam desain *interface* untuk memberikan estetika yang menyenangkan dalam memfasilitasi fungsi pembelajaran dan fungsi panduan. Dimana, tercapai dengan mengaplikasikan elemen dari grafis dan warna ke dalam desain *interface*. Dalam hal ini, ini difasilitasi melalui penggunaan dari ruang yang terfasilitasi dengan jelas teridentifikasi bahwa harmoni dalam suara suatu *interface* mendukung dan dapat memotivasi, memperjelas pesan, dan konsentrasi dari user. Bila disimpulkan, ada 38 karakteristik dari 6 elemen desain *interface*, seperti dengan teks, warna, gambar atau grafik, animasi, video, dan suara yang telah diidentifikasi. Karakteristik dari setiap elemen telah diidentifikasi sesuai dengan karakteristik dari elemen teks yang memiliki karakteristik terbanyak

### © Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Institut Bisnis dan Informatika Kwik Kian Gie

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



dan elemen suara yang terdikit. 10 karakteristik untuk elemen teks telah diidentifikasi, diikuti 6 karakteristik untuk warna, 7 karakteristik untuk gambar atau grafik, 7 karakteristik untuk animasi, dan 6 karakteristik untuk video dan suara. Dari hasil ini, didapatkan hasil bahwa kebanyakan elemen memiliki ciri yang mirip dengan prinsip *guidelines* dalam aplikasinya. Ini dapat dilihat, seperti dalam prinsip konsistensi dimana elemen teks, warna, grafis, dan animasi membutuhkan penggunaan yang konsisten dari jenis font yang sama, penyelarasan, ukuran, warna, warna latar, grafis, posisi animasi, dan grafis sepanjang *interface*. Karakteristik dari elemen desain *interface* diidentifikasi dari *contextual review* sepanjang kelima prinsip yang sering diidentifikasi dari desain *interface* yang dijelaskan dalam tabel berikut.

Prinsip	Elemen-elemen	Karakteristik
Konsisten Hirarki Kontras Keseimbangan Harmoni	Teks	Font, ukuran, peletakan, dan warna yang konsisten Paragraf pendek, menghindari semua huruf cetak besar san serif untuk <i>copy</i> badan, tidak menggunakan <i>decorative</i> , script, atau tulisan hitam Kontras dengan latar belakang <i>Cluster</i> , terstruktur, dan hierarkis Paragraf dalam baris kolom, rata kiri Menggunakan huruf cetak kecil, membatasi panjang garis Animasi garis bawah untuk menarik perhatian Membatasi jumlah jenis font Integrasi teks dan diagram sepantasnya
	Warna	Konsisten sepanjang halaman Memperhatikan kultur, asosiasi kolom Kontras, latar depan, dan latar belakang Konten kelompok, hierarkis, dna fungsional Menghindari warna cerah untuk teks panjang

**© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)**

Hak Cipta Dilindungi Undang-Undang

**Institut Bisnis dan Informatika Kwik Kian Gie**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

	Enak dipandang, memukau
Grafis atau gambar	Konsisten sepanjang halaman Fungsional teks dan diagram sepantasnya konten yang pantas, mudah dimengerti <i>Legibility</i> , hirarkis, menekankan, kontras, harmonis Jelas, menghindari kontras berantakan untuk kemudahan baca Menghindari gambar sebagai latar belakang Mengkonsiderasi tingkat hadirin
Animasi	Konsisten sepanjang halaman identifikasi <i>user</i> , kebutuhan, dan fungsi Untuk perhatian, pengertian, dan pengingatan Tidak mengganggu konten lain Simpel, minimal, tidak terlalu cepat Enak dipandang, memukau
Video	Peletakan yang terlihat, penggunaan minimal Biasa untuk pengenalan Pemecahan video per subtopik Kontras, kejelasan pada latar belakang Simpel, dapat dimengerti, ditemani suara Menggabungkan <i>user control button</i>
Suara	Konsisten sepanjang ada Enak, koheren, konsisten Cocok dengan konten dan suasana, sinkron

Tabel 2 1 Prinsip dari Desain Interface

Sumber: ((Kamaruddin and Sulaiman, 2018))

## B. Bahasa pemrograman alamat situs

### 1. No Code

*No-Code* adalah sebuah perangkat, yang mempermudah user tanpa kemampuan teknis untuk membuat sebuah perangkat lunak tanpa membuat suatu baris kode. Alat ini biasa disertai dengan *interface* yang mempermudah



*user* untuk menggunakan fasilitas-fasilitas yang tersedia di perangkat yang digunakan. *Framework No-Code* adalah sebuah alat pemrograman yang menggunakan *interface* pengembangan visual untuk mempermudah user dalam pembuatan aplikasi dengan komponen *drag and drop*. (<https://kissflow.com/low-code/no-code/no-code-overview/#what-is>, 2021, diakses bulan Agustus 2021)

Pengembangan dengan menggunakan *No-Code* memiliki karakteristik sebagai berikut:

- a. *Interface drag and drop* untuk kemudahan proses pembuatan aplikasi
- b. Model visual untuk membuat *User Interface*, model data, dan fungsionalitas dengan opsi untuk menambahkan coding teks ketika diperlukan.
- c. Sebuah struktur untuk mengolah *database*.
- d. Fungsionalitas untuk membuat kode baru, tanpa harus membuat modul dari nol.
- e. Pengelolaan aplikasi, untuk membangun, menyebarkan, debug, dan proses staging serta produksi.

## 2. *Bubble*

*Bubble* adalah sebuah alat untuk membuat sebuah situs web tanpa coding, dengan tujuan memberi kuasa kepada entrepreneurs untuk membuat *production-ready* aplikasi web. *Bubble* bertujuan untuk memberi nilai jual dan berkembang sesuai dengan user. Sebagai perusahaan yang didukung oleh investor, *Bubble* tetap berdedikasi kepada *value driven mindset*. Tujuan dari *Bubble* adalah membangun *platform* terbaik untuk memberi *user* kemampuan



membuat situs web tanpa menulis kode. (<https://bubble.io/faq> & <https://bubble.io/features>, 2021, diakses bulan Agustus 2021)

Bubble memiliki kemampuan untuk membuat situs web tanpa koding, dengan beberapa elemen sebagai berikut:

**a. Develop**

*Bubble* memberi kita kuasa untuk membuat secara interaktif, - *multi-user apps* untuk *desktop* dan situs web, serta memiliki semua alat yang dibutuhkan untuk membuat sebuah situs seperti *Facebook* atau *Airbnb*. Fitur yang diberikan untuk pengembangan ini berupa:

1. *Kostumisasi User Experience*

Dalam fitur ini diberi kemampuan untuk membuat sebuah logik dan mendeterminasikan apa yang terjadi ketika user melakukan navigasi ke dalam aplikasi.

2. *Pengolahan data dan akun*

Desain, pencarian, dan ekspor struktur data dalam storage data *Bubble*. Memberi kemampuan kepada pengguna untuk membuat akun user, yang kompatibel dengan *log-in* melalui *Facebook*, *LinkedIn*, atau *Google*. Setiap aplikasi *Bubble* diberi *ready-made* manajemen sistem user.

3. *Integrasi dengan segalanya*

Kembangkan aplikasi *Bubble*-mu dengan *plugin Javascript*, atau melakukan koneksi dengan *API (Application Programming Interface)* dengan *API connector Bubble*.

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Institut Bisnis dan Informatika Kwik Kian Gie

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

## b. Desain

Membuat sebuah desain yang ramah terhadap tampilan *mobile* dan konten dinamis untuk memberi sentuhan kilap kepada produk yang dibuat, dengan fitur-fitur:

### 1. *Drag and Drop*

Membuat sebuah desain yang *pixel-perfect* dan memasukan gambar, ikon, video, map, dan seterusnya.

### 2. Penggunaan konten dinamis

Melakukan *display* konten yang dibuat oleh *user* atau data yang ditarik dari servis lain seperti *Stripe*, *Google*, dan *Facebook*. Ubah font, warna, atau visibilitas dari apa yang ada di halaman berdasarkan informasi yang diberi aplikasi pembuat dengan elemen dinamis Bubble.

### 3. Membuat aplikasi multibahasa

Mempermudah translasi bahasa secara otomatis ketika menggunakan Bubble, karena Bubble mendukung bahasa dari 80+ negara.

## c. *Host*

Dengan elemen ini, pembuat yang menggunakan Bubble tidak perlu mengkhawatirkan perawatan *server*, infrastruktur, atau operasi.

### 1. *Scale your application*

Dengan Bubble, pengguna dapat mengalokasikan server sesuai dengan kebutuhan dari yang situs web yang dibuat.

### 2. Kontrol versi dan *backups*





1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

Bubble menyediakan kemudahan, yaitu ketika melakukan testing ataupun perubahan ketika situs web telah di-*deploy*, Bubble menyediakan *backup* sehingga revisi terhadap apa yang telah di-*deploy* dapat dikembalikan.

3. Tetap aman dan privat

Bubble memiliki sertifikasi *Secure Sockets Layer (SSL)* untuk semua situs yang terbuat di *Bubble*.

### 3. *Hypertext Preprocessor (PHP)*

*PHP* adalah bahasa yang kita gunakan untuk membuat server menghasilkan output dinamis, dengan potensi berbeda-beda setiap kali situs web memberi *request*. Secara umum, dokumen *PHP* berakhir dengan *.php*. Ketika sebuah situs web menemui ekstensi tersebut, secara otomatis melakukan passing ke *processor PHP*. Tentu saja, server *web* mudah dikonfigurasi, dan beberapa developer web memilih untuk memaksan file berakhir dengan *.htm* atau *.html (HyperText Markup Language)* untuk mendapatkan parsing dari *processor PHP*, biasa dikarenakan mereka ingin menutupi fungsi *PHP* mereka. (Nixon, 2018)

*PHP* bertanggung jawab untuk membalikan file bersih yang cocok untuk display dalam situs web. Dalam simplenya, dokumen *PHP* akan memberikan output hanya *HTML*. Untuk membuktikan ini, kalian dapat mengambil dokumen *HTML*, dan menyimpannya sebagai dokumen *PHP* (dengan contoh, menyimpan *index.html* sebagai *index.php*) dan mendisplaynya sesuai dengan original.

*PHP* memiliki banyak struktur yang digunakan, dengan berikut:

#### 1. Menggunakan Komen



Dimana fitur ini berguna untuk sementara menghilangkan kode dari program, dengan tujuan mencari tahu apakah error atau tidak. Contoh:

```
// echo "X equals $x";
```

## 2. Sintaks Dasar

*PHP* cukup simpel, dengan akar dari *C* dan *Perl*, tetapi mirip juga dengan *java*. Ini membuat *PHP* sangatlah fleksibel, dengan sintaks seperti berikut:

- *Semicolon*

```
$x += 10;
```

- *Simbol\$*

```
$mycounter =1;
```

## 3. Variabel

Ada sebuah metafor yang bisa membantu, dimana variabel adalah sebuah korek yang kita beri nama dan kita cat, dengan variabel berikut:

- *String Variables*

- *Numeric Variables*

- *Arrays*

- *Two-dimensional arrays*

- *Variable-naming rules*

## 4. Operator

Operator mengizinkan anda untuk membuat sebuah formula matematika untuk dilakukan, seperti dengan adisi, subtraksi, multiplikasi, dan divisi. Dengan beberapa operator berikut:



- *Arithmetic operators*
- *Assignment operators*
- *Comparison operators*
- *Logical Operators*

## 5. Tugas Variabel

Sintaks untuk memberikan nilai variabel, dimana selalu variabel = nilai, atau untuk memberikan nilai baru kepada variabel baru. Dimana dengan sintaks-sintaks berikut:

- *Variabel incrementing and decrementing*

```
[ ++$x ]; [ --$y; ]
```

- *String concatenation*

```
echo "You have " . $msgs . " messages.";
```

- *String types*

```
$info = 'Preface variables with a $ like this: $variable';
```

- *Escaping characters*

```
$text = 'My spelling's atrocious'; // Erroneous syntax
```

## 6. Multiple-Line Command

Ada waktu dimana dibutuhkannya output dengan teks dari *PHP*, dan menggunakan beberapa command *echo* atau *print* akan memakan waktu cukup banyak. Contohnya:

```
<?php $author = "Steve Ballmer";
```

```
echo "Developers, developers, developers, developers,  
developers, developers, developers, developers, developers! -  
$author."; ?>
```

## 7. Penelitian Variabel

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



*PHP* adalah bahasa yang cukup longgar. Ini menandakan bahwa variabel tidak melakukan deklarasi sebelum digunakan, dan *PHP* akan mengkonversi variabel ke tipe yang dibutuhkan dari konteks ketika diakses. Contohnya:

```
<?php
$number = 12345 * 67890;

echo substr($number, 3, 1); ?>
```

### 8. Konstan

Konstan mirip dengan variabel, menyimpan informasi yang akan diakses nanti, kecuali mereka seperti dengan apa yang disebutkan – konstan. Dengan konstan sebagai berikut:

```
$directory = ROOT_LOCATION;
```

### 9. Konstan yang ditentukan

*PHP* datang dengan beberapa konstan yang telah disediakan, yang tidak sering digunakan pemula. Tetapi, ada yang dikenal dengan *magic constants*, yang akan ditemui berguna.

__LINE__	jumlah baris angka di file
__FILE__	jalan lengkap dari file dan nama file
__DIR__	direktori dari file
__FUNCTION__	nama dari fungsi
__CLASS__	nama class
__METHOD__	nama dari method class
__NAMESPACE__	nama cari namespace sekarang

### 10. Perbedaan *echo* dan *print*

Kedua *command* tersebut mirip, tetapi *print* adalah fungsi seperti konstruk yang mengambil satu parameter dan memiliki nilai balik (selalu satu), dimana *echo* secara murni konstruk bahasa *PHP*.

### 11. Fungsi

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



Fungsi yang membagi *sections* dari kode yang menjalankan tugas spesifik. Contoh, terkadang dibutuhkan mencari sebuah tanggal dan membalikannya dengan format tertentu.

## 12. *Scope Variabel*

Bila program yang dibuat panjang, cukup mungkin akan kekurangan nama variabel yang bagus, tetapi dengan *PHP* dapat menentukan *scope* variabel. Dalam kata lain, kita dapat dengan contoh, bila kita ingin variabel \$temp digunakan dalam fungsi tertentu, dan melupakannya ketika fungsi dikembalikan.

## 4. *My Structured Query Language (MySQL)*

*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *Structured Query Language (SQL)* atau *Database Management System (DBMS)* multi alur, multi pengguna (Nixon, 2018). *MySQL* merupakan implementasi sistem *Relational database management system (RDMS)* yang didistribusi secara gratis dibawah lisensi *General Public License (GPL)*. *MySQL* memiliki kapabilitas seperti (Nixon, 2018) :

1. Kemampuan prose *query* sederhana, dalam arti dapat memproses *SQL* per satuan waktu
2. Memiliki banyak tipe data
3. Mendukung field yang Diberi Index
4. *MySQL* memiliki beberapa lapisan keamanan, seperti *subnet mask*, yang telah dienkripsi
5. *MySQL* dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP



6. *MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik
7. *MySQL* memiliki fungsi dan operator secara penuh yang mendukung *select* dan *where*
8. *MySQL* memiliki struktur tabel fleksibel
9. *MySQL* dapat menampung basis data dalam skala besar
10. Mendukung penuh terhadap kalimat *SQL*, *GROUP BY* dan *ORDER BY*. Mendukung juga fungsi penuh [ *COUNT()*, *DISTINCT()*, *AVG()*, *STD()*, *SUM()*, *MAX()*, dan *MIN()* ]

### C

#### Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

#### C. Hosting

*Hosting*, atau *Web Hosting* adalah sebuah servis yang memungkinkan individu atau kelompok untuk membuat online situs web milik mereka. *Hosting* situs web atau penyedia servis *hosting*, adalah sebuah bisnis dengan tujuan memberi servis untuk membuat online situs web, sehingga ketika user ingin membuka situs web yang telah dibuat, user dapat melakukan akses secara langsung ke situs web yang telah di *hosting*. (Dudhane *et al.*, 2020)

#### D. Multi Criteria Decision Making (MCDM)

*MCDM* banyak digunakan dalam lingkup luas, dikarenakan *MCDM* mempermudah pengambilan keputusan yang sulit diukur secara pasti, karena suatu masalah memiliki kerumitan dan kompleksitas yang berbeda-beda, sehingga pengambilan keputusan yang pasti tidak selalu dapat dilakukan.

*MCDM* merupakan pendekatan atau metode dalam pengambilan keputusan berdasarkan alternatif solusi dari multikriteria. Pada dasarnya *MCDM* merupakan bagian dari bidang keilmuan riset operasi (Rachman, Eka, Yusriana dan Rizki 2020). Secara teknis, *MCDM* melibatkan kriteria, aktor, dan tujuan. Dimana ada 5 aspek

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



dalam *MCDM*, yaitu tujuan, preferensi pengambilan keputusan, alternatif, kriteria dan manfaat.

**E. Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)**

*Topsis* adalah sebuah metode yang menggunakan pemasukan minimal dari user dan mengeluarkan informasi yang mudah dimengerti. Ide dasar dari *Topsis*, adalah solusi yang paling optimal merupakan solusi yang memiliki jarak terdekat ke solusi ideal dan jarak terjauh merupakan solusi yang paling tidak ideal.

*Topsis* adalah sebuah metode klasikal berbasis alat optimasi *MCDM*. *Topsis* sudah dirancang untuk menyelesaikan permasalahan pemilihan jalan tengah atau pencapaian solusi. Teknik ini digunakan karena bersifat unik serta mudah dimengerti, sehingga disukai beberapa orang. *Topsis* terdiri dari dua solusi alternatif, yang terbaik dan yang terburuk. *Topsis* secara bersamaan, melakukan perhitungan untuk menentukan jarak untuk solusi terbaik, bersama dengan solusi terburuk. (Khan and Maity, 2017)

Pendekatan ini dapat dikategorikan sebagai berikut:

1. Evaluasi alternatif sesuai dengan tiap kriteria,  $x_{ij}$ , dimana  $i: (1 \leq i \leq m)$  menjelaskan alternatif dan  $j: (1 \leq j \leq n)$  menjelaskan kriteria, menghasilkan nilai  $m \times n$  dalam evaluasi matriks  $(x_{ij})_{m \times n}$ .
2. Berat kriteria  $w_j$ ; umumnya,  $\sum w_j = 1$ .

Dapat didetailkan sebagai berikut:

$$X = \begin{matrix} A_1 \\ \vdots \\ A_m \end{matrix} \begin{pmatrix} C_1 & \dots & C_n \\ x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \\ w_1 & \dots & w_n \end{pmatrix}$$

Dimana, pendekatan ini memiliki 6 tahapan, sebagai berikut:

Hak Cipta Dilindungi Undang-Undang

Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Institut Bisnis dan Informatika Kwik Kian Gie

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



- ◆ Tahap 1: Membuat matriks normalisasi

Ⓒ Untuk mempermudah perbandingan, tahap ini bertujuan untuk menghasilkan data sebagai berikut:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}},$$

Dimana  $r_{ij}$  merepresentasikan hasil normalisasi alternatif  $i$  sesuai kriteria  $j$ .

- Tahap 2 : Konstruksi matriks terbobot

Menggunakan kriteria berat  $w_j$ , dilakukannya komputasi berat normalisasi

evaluasi  $v_{ij}$  sebagai  $v_{ij} = w_j \times r_{ij}$ .

- Tahap 3: Menentukan solusi ideal

Rumus untuk solusi ideal adalah:

$$A^* = \{v_1^*, \dots, v_n^*\} = \{(\max_i v_{ij} | j \in \Omega_b), (\min_i v_{ij} | j \in \Omega_c)\}$$

Rumus untuk solusi tidak ideal adalah:

$$A^- = \{v_1^-, \dots, v_n^-\} = \{(\min_i v_{ij} | j \in \Omega_b), (\max_i v_{ij} | j \in \Omega_c)\}$$

Dimana, ketika  $\Omega_b$  adalah kumpulan kriteria dan  $\Omega_c$  adalah kumpulan *cost* kriteria.

- Tahap 4: Perhitungan jarak tiap alternatif dan tiap solusi

Rumus untuk jarak solusi ideal adalah:

$$D_i^* = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^*)^2}, i = 1, \dots, m,$$

Rumus untuk jarak solusi tidak ideal adalah:

$$D_i^- = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^-)^2}, i = 1, \dots, m$$

- Tahap 5: Perhitungan jarak relatif terdekat dengan solusi ideal





$$CC_i = \frac{D_i^-}{D_i^- + D_i^+}, \quad i = 1, \dots, m.$$

- Tahap 6: Menilai berdasarkan nilai  $CC_i$ .

Dimana, semakin dekat  $CC_i$  dengan 1, maka semakin tinggal nilai alternatif tersebut untuk diberi penilaian. (Alaoui, Mohamed El, 2021)

## F. Unified Modeling Language (UML)

*UML*, adalah sebuah bahasa grafis yang secara resmi didefinisikan sebagai *Object Management Group (OMG)* untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi artefak dari sebuah perangkat lunak sistem. Artefak dalam artian ini, adalah sebuah benda yang dibuat atau dikoleksi dalam proses pembuatan ( contoh dari artefak, memasukan penggunaan *use case*, *requirements*, *design*, *code*, *executable files*, dan lain-lain). *UML* memberikan sebuah cara standar untuk menulis sebuah rancangan sistem, termasuk dengan hal-hal konseptual seperti dengan proses bisnis dan fungsi sistem, bersama juga dengan hal padat seperti *statement* bahasa pemrograman, skema *database*, dan perangkat lunak yang dapat digunakan kembali. (Osis and Donins, 2017)

### 1. Diagram Use Case

Use Case Diagram adalah sebuah diagram yang menunjukkan kumpulan dari beberapa *use case*, *actors* dan hubungan antaranya. Diagram *Use Case* biasanya digunakan untuk mengorganisasi dan pembuatan model kegiatan yang dibutuhkan dari sebuah sistem.

### 2. Diagram Aktivitas

Diagram aktivitas adalah diagram yang menampilkan model dinamis dari sistem yang berisikan urutan skema dari aktivitas pertama menuju

aktivitas selanjutnya. Sebuah aktivitas menunjukkan kumpulan tindakan yang berurutan atau bercabangan dari skema. Diagram aktivitas biasanya digunakan oleh organisasi dalam menggambarkan model bisnis, proses *engineering* dan alur kerja.

### 3. *Class Diagram*

*Class diagram* adalah diagram yang paling sering ditemukan pada sistem yang berbasis objek dan digunakan untuk menggambarkan sudut pandang dari sebuah sistem. *Class diagram* menunjukkan kumpulan kelas antarmuka dan hubungan antar kedua hal tersebut.

## **© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)**

Hak Cipta Dilindungi Undang-Undang

**Institut Bisnis dan Informatika Kwik Kian Gie**



**KWIK KIAN GIE**  
SCHOOL OF BUSINESS

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.